Efficient and Optimal Storage Placement in Sensor Network

Mr.Sagar M Mane **Mr.Vishal V Bhanwase** **Mr.Avadhoot S Joshi**

Abstract ---In sensor network a large amount of data need to be collected for future information retrieval. The data centricstorage has become an important issue in sensor network.Storage nodes are used in this paper to store and processthe collected data. This paper considers the storage node placement problem aiming to place unlimited storage nodes in sensor network to minimize the total energy cost for collecting the raw data and replying queries at the storage nodes. In this paper a strong data access model for placing storage nodes in sensor network is presented.We consider an application in which sensor networksprovide real time data services to user. The main aim of this paper is to reduce the cost for raw data transfer,query diffusion, query reply by defining the best locationof storage nodes in sensor network.

Index Terms--Wireless sensor network, data query, data storage, data reply, query diffusion.

1. INTRODUCTION

One of the key challenges in wireless sensor network is the storage and querying of useful sensor data. The wireless sensor network is built of nodes, where each node is connected to one or more sensors. Sensor networks deployed for different computing applications,e.g.,sensing environmental or earth condition and monitoring people's behaviors, generates a large amount of data over a long period of time. Storage is an essential factor of any data centric sensor network application. One of the main challenges in these applications is how to search and store the collected data. The collected data can either be stored in the network sensors, or transmitted back to the sink and stored there for future retrieval. This design is ideal since data are stored in a central place for permanent access. Placing unlimited storage nodes is related to the sensor network. Query is the most important part of sensor network since in aspect sensor network provides the information about environmental condition to the end user.Therefore, we aim to minimize the total energy cost and data query by accurately deploying the storage nodes in sensor network. In section 4 we discuss the data access model. In section 5, we present the conclusion and future work.

2. RELATED WORK

There has been a lot of prior research work on data querying models in sensor network. In early models [1, 2, 3], query is spread to every sensor node by flooding messages. Sensors nodes send data back to the sink in the reverse direction of query messages. Those methods do not consider the storage concern in sensor networks.

PRESTO [4] is a recent research works on storage architecture for wireless sensor networks. A proxy layer is introduced between sensor nodes and user terminals and proxy nodes can cache previous query responses. When a query arrives in a proxy node, it first checks if the cached data can satisfy the query before forwarding the query to other nodes. Compared with the storage nodes in this paper, Nodes in PRESTO have no resource constraints in term of computation, power, storage and communication. It is a more familiar storage architecture that does not take the characteristics of data generation or query into consideration.
Data-centric storage schemes [5, 6, 7] store data to different places in sensor networks according to different data types. In [6, 7], the authors propose a data-centric storage scheme for sensor networks, which inherits ideas from distributed hash table. The home site of a data is obtained by applying a hash function on the data type.
LEACH [8] is a clustering based routing protocol, in which cluster heads can fuse the data collected from its neighbors to reduce communication cost to the sink. LEACH has a similar structure to our scheme, having cluster heads aggregate and forward data to the sink. However, LEACH aims to reduce data transmission by aggregating data; it does not address storage problem in sensor networks

3. Placing Unlimited Number of Storage nodes

Algorithm1 finds the optimal placement of storage node for the case ßRq = Rd.Assume that n nodes in the tree T are labeled using the post order. A table e*[1..n] is used to hold the minimum energy cost of all sub trees rooted at node i = 1, . . . , n. So at the end of the computation, e* [n] will hold the minimum energy cost of T. we also maintain a second table ef[1..n] which records the energy cost of all sub trees when all nodes in each sub tree are forwarding nodes. In the algorithm, line 5-9 compute the e* and ef entries for all leaves and lines 10-19 compute the e* and ef entries for the remaining nodes

Let i be any node in the communication tree and T subtree rooted at i. We use |Ti| to denote the number of Nodes in T I . We define E(i) to be the energy cost incurred at i per time unit, which consist, the cost for raw data transfer i to its parent if i is a forwarding node, the cost for query diffusion if i has storage nodes as its descendants, and the cost for query reply if i is a storage node or has a storage descendant. To define E(i) mathematically we need to consider several possible cases.

```
1: make the root a storage node
2: if ßRq = Rd then
3: make all non-root nodes forwarding nodes and return
4: end if
5: for all leaves i do
6::make i a storage node
7: e*[i]=RqßSd
8: ef[i]=RdSd
9: end for
10: for all remaining nodes i do
11: make i a storage node
12: min1=Rqß|Ti|Sd+BiRqSq+∑
 e*[j]
13: min2=Rqß|Ti|Sd+∑
 ef[j]
14: e*[i]=min{min1,min2}
15: ef[i]=|Ti|RdSd+∑j€cij€cij€ci ef[j]
16: if min1=min2 then
17: change each descendent of i that is a storage node
    to a forwarding node
18: end if
19: end for
```

Algorithm.1: Placing unlimited storage nodes

Case I: i is a forwarding node and there are no storage nodes in Ti. All raw data generated by the nodes in T be forwarded to the parent of i and there is no query diffusion cost. So E(i) = |Ti|RdSd .

Case II: i is a storage node and there are no other storage nodes in Ti. The latest readings of all raw data generated by the nodes in Ti are processed at node i and the reduced reply size will be ß|Ti|sd.Node i sends the reply to its parent when queries arrive. So E(i) = Rq ß|Ti.|Sd

Case III. i is a storage node and there is at least one other storage node in Ti. In addition to the cost for query reply as defined in Case II, i also incurs a cost for query diffusion that is implemented by broadcasting to its children. So E(i) = Rqß|Ti|Sd+ BiRqSq.

Our algorithm relies on following lemma.

Lemma 1.Given a node i and its sub tree Ti. If ßRq=Rd, then I must be a forwarding node to minimize e(i).if ßRq=Rd, then I must be a storage node to minimize e(i).

Proof: First we compare the two trees based on theirenergy cost, which are equivalent in every aspect except that thefirst tree's root is a forwarding node and the second tree's root isa storage node. Let e1 and e2 be the two trees based on theirenergy cost. Comparing the two trees using the energy cost ofindividual nodes, one by one, we observe that any two non- rootnodes in the same position of the trees must have the similarenergy cost. The only change is the energy cost of the roots.

Let E1and E2be the energy cost of the roots in the two trees, respectively. Therefore, e1- e2= E

We consider two cases. First, if both root have no storage descendants, then according to the four different case definition of energy cost (case I and II), we have

E1-E2 = |Ti| Rd Sd -Rqß |Ti|Sd
= Ti| Sd(Rd-ßRq)

Second, if both roots have at least one storage descendant, then according to the four different case definition of energy cost (Cases III and IV), we have

E1-E2 =((d1+1)RdSd+BiRqSq+Rqßd2Sd) –(Rqß|Ti|Sd+BiRqSq)
= (d1+1)Sd(Rd-ßRq)

4. System Architecture

In this paper, we consider an application in which sensor networks provide real-time data services to users. A sensor network is given with one defined sensor identified as the sink (or base station), access point and many normal sensors, each of which generates (or collects) data from its environment. Users or application program specify the data they need by submitting queries to the sink and they are usually interested in the latest readings generated by the sensors. To reply to queries, one typical solution, shown in fig.1, is the sinks have all the data.

.

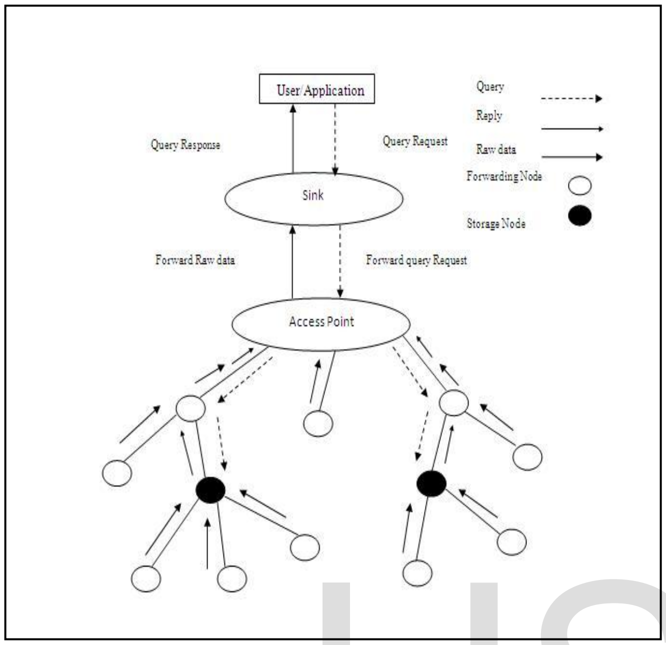


Figure1: Data Access Model

This requires each sensor to send its readings back to the access point immediately every time it produces new data. Transferring all raw data could be very expensive and is not always required. Alternatively, we allow sensors node to hold their raw data and to be aware of the different queries, then raw data can be managed to contain only the readings that users are interested in and the reduced reply size, instead of the whole raw data readings, can be send back to the sink. This design is illustrated in Fig. 1, where the black sensor nodes, called storage nodes, are allowed to hold raw data. The base station diffuses queries to the access point by broadcasting to the sensor network and then access point broadcast the queries to storage sensors and these storage sensors reply to the queries by sending the processed data back to the storage node. Compared to the earlier solution, this approach reduces cost of the raw data transfer because some raw data transmissions are replaced by query reply. On the other hand, this scheme incurs an extra query diffusion cost (as figured by the dashed arrows). In this paper, we are interested in vital designing a data access model to minimize energy cost associated with query diffusion, raw data transfers, and query replies.

Access Point: When the user fires the query on the sink, sink forward the query Request to the access point. Access point broadcast the query to sensor nodes. When the query arrived at storage nodes they forward the raw data back to the access point and then access point obtain the result and forward the data to the sink..
We first formally define two types of sensors (or nodes):

Storage nodes: These types of nodes have much larger storage capacity than normal sensor nodes. In the data access model as shown in fig.1, they store all the data received from other nodes or generated by themselves. Storage node does not send anything until queries arrive. According to the query specification, they receive the results needed from the raw data they are holding and then return the results back to the base station. The base station itself is considered as a storage node.

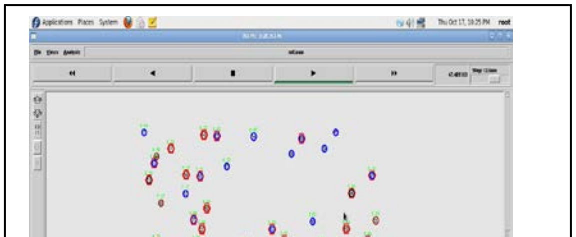
Forwarding nodes: These types of nodes are regular sensors and they always forward the data received from other nodes or generated by themselves along a path towards the sink. The outgoing data are kept intact and the forwarding operation continues until the data reach a nearest storage node. The raw data forwarding operation is independent of queries and there is no data processing at forwarding nodes.

Rd:-Data generation rate
Sd : Size of each data
c_i : the number of node i's children
Rq: User query rate
Sq : size of each query message
n: total number of sensors
e(i):Energy cost of all the nodes in Ti.
E(i): energy cost of node i
β: Data reduction rate
T_i : the subtree rooted at node i

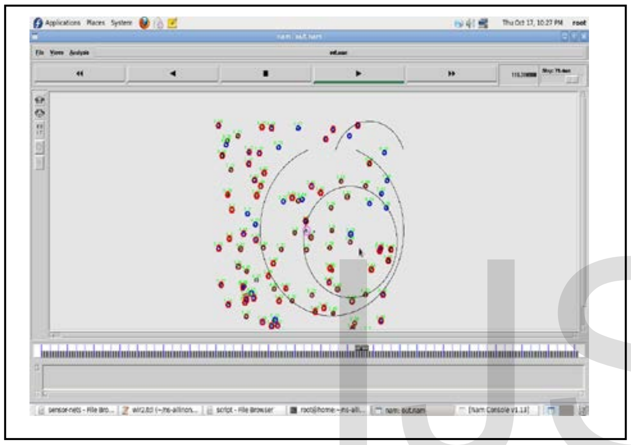
Table 1: Notations

5. Simulation Settings

In this paper, the communication among all n nodes is based on a tree topology with the sink as the root. The tree is formed in the initial phase as follows. The sink first broadcasts a Message with a hop counter. The nodes receiving the message will set the message sender as the parent node, increase the hop counter by one, and broadcast it to their neighbors.
To transmit one data unit, the energy cost of the sender and receiver are etr and ere respectively. etr is also relevant to the distance between the sender and receiver. This etr and ere only we update, when we receive the energy cost for transmitter and receiver.

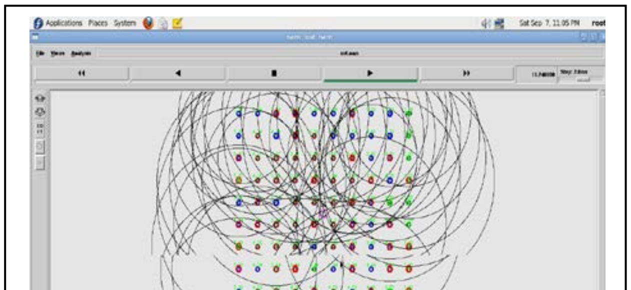


To simplify the problem, we set the length of each tree edge to one unit, which means that sensor nodes have a fixed transmission range and the energy cost of transferring data is only proportional to the data size.
Our algorithm assumes that the energy cost is proportional to the transmitted data size, which can be realized in many communication protocols such as the duty cycle mechanism.



In some duty cycle mechanism schemes, the network delay can be reduced by using carefully designed transmission schedule plan. Moreover, we assume that the applications considered in this project can tolerate the delay caused by low layer communication, such as retransmission and duty cycle mechanism. So we need to compute duty cycle.

In our simulation settings, we consider a network of sensor deployed on a disk of radius 5 with the sink placed at the center. One hundred and one sensor nodes (n=100) are deployed to the field randomly. After all nodes are deployed in a routing tree rooted at the sink and is constructed by broadcasting a message from the sink to all the nodes in the network. Therefore, for a certain set of parameters, we conduct several independent trials .We set the following parameters in our simulations: rd=0.5, rq=0.4, sd=512, sq=512.We evaluate the energy cost by considering the varying number of storage nodes and varying the data reduction rate α.



First we create the routing protocol and this routing is attached with every node using routing agent. When we create the node we call the routing protocol so each node will start and working as per the protocol procedure. In this scenario node 0 is acting as a base station. Next we create the wireless nodes this node is used to sense the application. In application layer we have attached the sensed application.

6. Resulted Graphs

Fig1. shows the comparison of Total Energy Cost with varying data reduction rate. There are three graphs in fig. The red graph shows the energy cost with varying data reduction rate in unlimited storage node, the green one deal with the energy cost with varying data reduction rate in limited storage node and the blue graph shows the energy cost with varying data reduction rate in c ary regular tree.

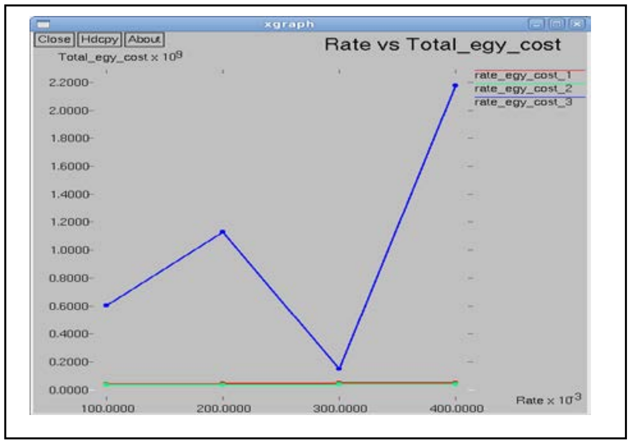


Figure 2. Comparison of Total Energy cost with varying data reduction rate.

Fig.2 shows the comparison of Time with varying packet delivery. There are three graphs in above fig. The red graph shows the Time with packet delivery in unlimited storage node, the green one deal with the Time and corresponding packet delivery in limited storage node and the blue graph shows the Time and corresponding packet delivery in c ary regular tree.

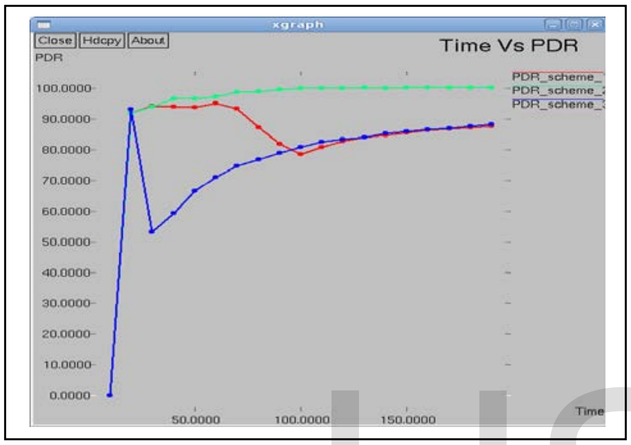


Figure 3.Comparison of Time with varying packet delivery

Fig.3 shows the comparison of Time with normalized routing overhead. There are three graphs in above fig. The red graph shows the Time with normalized routing overhead in unlimited storage node, the green one deal with the Time and corresponding normalized routing overhead in limited storage node and the blue graph shows the Time and corresponding normalized routing overhead in c ary regular tree.

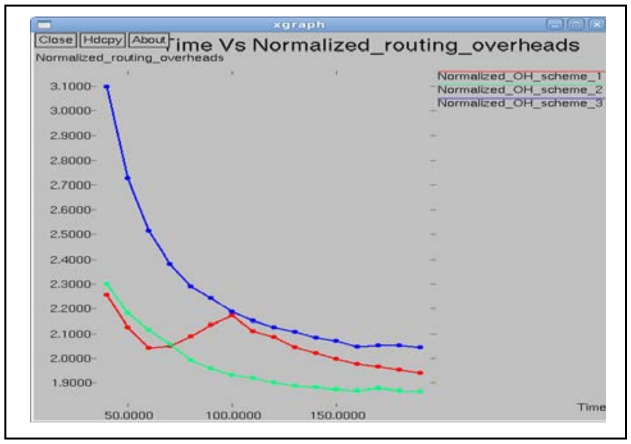


Figure 3. Comparison of Time with normalized routing overheads.

Fig.4 shows the comparison of Time with Delay. There are three graphs in above fig. The red graph shows the Time with Delay in unlimited storage node, the green one deal with the Time and corresponding delay in limited storage node and the blue graph shows the Time and corresponding delay in c ary regular tree.

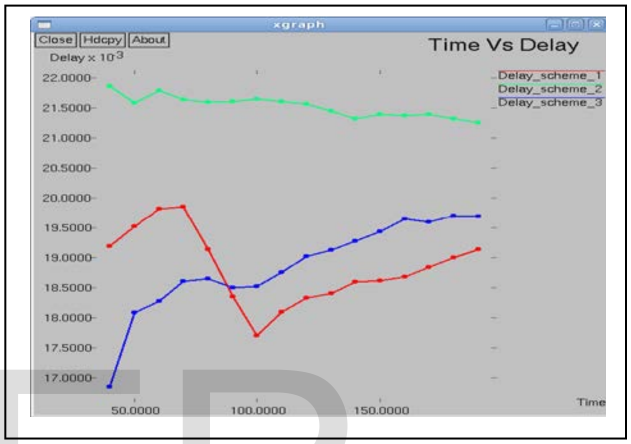


Figure 4. Comparison of Time with Delay

7. Conclusion

This paper considers the storage node placement problem in a sensor network. This paper introduces unlimited number of storage nodes in sensor network release the cost of sending all the raw data to a central place. In this paper, first examine how to place unlimited number of storage nodes to save energy for data collection and data query. This new model is much more simplified and implementable. We have tested it on different data sets available on internet using network simulator software. Our Future work includes Evaluation of limited number of storage nodes in sensor network to optimize query reply in a sensor network and to solve the storage node placement problem in terms of other performance metrics.

8. References

[1] Sagar M Mane, Prof. Dr. S .S. Apte, “Evaluation of unlimited storage: Towards better data access model for “Sensor Network”,” in International Journal of Computer

Applications (0975-8887) Volume 73/Number 7 (ISBN: 973-93-80876-07-5)

[2] Prof. Dr. S .S. Apte, Sagar M Mane, "Improvement of limited storage placement in wireless sensor network" in International Organization of Scientific Research Volume 12,Issue 2(May-Jun. 2013),PP 107-111 (e-ISSN:2278-0661,p-ISSN:2278-8727).

[3] Sagar M Mane, Dr. Mrs. S. S. Apte, "Modified Storage Placement in Wireless Sensor Network" in International Journal of Engineering Research & Technology Vol. 2 Issue 6, June – 2013 (ISSN: 2278-0181).

[4] C. Intanagonwiwat, R. Govindan, and D. Estrin. rected
diffusion: a scalable and robust Communication paradigm
for sensor networks. In Proceedings of the 6th annual international conference on Mobile computing and networking, pages 56–67, New York, NY, USA, 2000. ACM Press

[5] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed diffusion for wireless sensor networking. IEEE/ACM Trans. Netw., 11(1):2– 16, 2003.

[6] S. Madden, M. J. Franklin, J. M. Hellerstein,and W.Hong TAG: a tiny aggregation service for ad-hoc sensor networks.SIGOPS Opererating System Review, 36(SI):131146,2002

[7] J. Newsome and D. Song. GEM: graph embedding for routing and data-centric storage in sensor networks without
geographic information. In Proceedings of the 1st international conference on Embedded networked sensor systems, pages 76–88, New York, NY, USA, 2003.ACM Press.

[8] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu. Data-centric storage in sensornets with GHT, a geographic hash table. Mobile Networks andApplications, 8(4):427–442, 2003.

[9] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D.Estrin. Data-centric storage in sensornets. SIGCOMM Computer Communication Review, 33(1):137–142, 2003.

[10] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan.Energy-efficient Communication Protocols for Wireless Microsensor Networks. In International Conference on System Sciences, Maui, HI, January 2000.

[11] P. Desnoyers, D. Ganesan, H. Li, M. Li, and P. Shenoy. PRESTO: A predictive storage architecture for sensor networks. In Tenth Workshop on Hot Topics in Operating System, 2005.

[12] P. Gupta and P. R. Kumar. The capacity of wireless networks IEEE Transactions on Information Theory, IT-46(2):388–404March 2000.

[13] D. Ganesan, D. Estrin, and J. Heidemann. Dimensions: why do we need a new data handling architecture for sensor
networks?SIGCOMM Comput. Commun. Rev., 33(1):143–148,2003.

[14] E. J. Duarte-Melo and M. Liu. Data-gathering wireless Sensor networks: organization and capacity . Computer Networks (COMNET), 43(4):519–537, Nov. 2003